REMARKS

These remarks follow the order of the paragraphs of the office action. Relevant portions of the office action are shown indented and italicized.

In particular, please note Claim 12 is amended to include the limitation stated in the response to arguments below. As such claim 12 should be allowable. Also claim 4 should be allowable, even if applicants other arguments are rejected.

If the Examiner chooses to allow any claim or claims, Examiner is authorized to cancel all the non-allowed claims in order to bring allowable subject matter to issue.

DETAILED ACTION

Response to Amendment

1. This office action has been issued in response to amendment file 10 April 2001. Claim 4 has been amended, Claims 1-20 are pending in this Office Action. Accordingly, this action has been made FINAL.

Specification

2. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1 .75(d)(1) and MPEP § 608.01(o).

Claims 5-8, 10, 14 recites the limitations of "a tangible computer readable medium'. However, "a tangible computer readable medium" is not specified in the specification. Correction of the following is required.

Claims 15, 18-20 recite the limitations of "a computer usable medium". However, "a computer usable medium" is not specified in the specification. Correction of the following is required.

Claim 17 recites the limitations of "a computer-readable recording medium". However, "a computer-readable recording medium" is not specified in the specification. Correction of the following is required.

3. Pending applicants' response to the objection supra, examiner will interpret "a computer readable medium", "a computer usable medium", "a computer-readable recording medium" to include only physical storage devices such as CD-ROM, magnetic disks, and to exclude medium for carrying signals and other forms of propagation or transmission medium.

In response, the applicants respectfully states that all the claims referred to above are amended as in the new listing of the claims to use the exact phrase already in the specification. The Examiner's interpretation is accepted. Thus, the specification need not be amended, and the objection is overcome.

Claim Rejections -35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

4. Claims 1-20 are rejected under 35 U.S.C. 102(a) as being anticipated by Peng et al. (XPath Queries on Streaming Data, International Conference on Management of Data, Proceedings of the 2003 ACM SIGMOD international conference on Management of data, published on June 9-12, 2003).

In response, the applicant respectfully state that applicants continue to maintain that Claims 1 - 20 are apparently not anticipated by the invention of Peng. Arguments previously made are repeated herein, except that claim 1, and the other independent claims were amended to make each narrower to bring this case to allowance quickly.

The present invention, claimed in Claims 1 - 20, provides:

"[A]n XML parser for inputting XML event strings which constitute an XML document to be processed, and an XPath evaluating unit for executing evaluation of the XPath by streaming processing are provided. This XPath evaluating unit serially evaluates the XPath with respect to the respective XML events transferred from the XML parser, and retains information concerning a result of partial evaluation of this XPath when the XPath

> is partially established for a given XML event. Then, when the last step of this XPath is established, the XPath is judged as established for the XML document.

Thus the present invention is concerned with XML parsing for inputting XML event strings. It serially evaluates the XPath with respect to the respective XML events transferred from the XML parser.

Whereas, the cited art to Peng, Article entitled: "Xpath Queries on Streaming Data". The Peng abstract reads:

> We present the design and implementation of the XSQ system for querying streaming XML data using XPath 1.0. Using a clean design based on a hierarchical arrangement of pushdown transducers augmented with buffers, XSQ supports features such as multiple predicates, closures, and aggregation. XSQ not only provides high throughput, but is also memory efficient: It buffers only data that must be buffered by any streaming XPath processor. We also present an empirical study of the performance characteristics of XPath features, as embodied by XSQ and several other systems.

Thus, Peng is concerned with design and implementation of the XSQ system for querying streaming XML data using XPath 1.0. Although Peng does parsing, evaluating etc., Peng is not concerned parsing when serially evaluating the XPath with respect to the respective XML events transferred from the XML parser. Examiner is apparently using personal knowledge to use and form elements into Peng that do not exist, automaton etc., that were not alluded to and were not a concern to Peng. Thus claims 1-20 are not anticipated by Pang, and are allowable.

The present response includes a copy of the referenced portions of Peng to more particularly show that Peng fails to teach or anticipate claims 1-20, especially as amended.

> ***With respect to claim 1,*** *Peng discloses an extensible-markup-language Path Language (XPath) evaluating method comprising evaluating the XPath relevant to an extensible-markup-language (XML) document by use of a computer, said step of evaluating being carried out individually concerning inputted XML events, while subjecting the XML document to streaming processing, step of evaluating (page 431 2nd paragraph of [1.]; page 432 fig. 1 example 1 and last paragraph), XPath method comprising:*

The cited Peng portion page 431 2nd paragraph of [1.] reads:

> We address the problem of evaluating XPath queries over streaming XML [23]. XPath is a well-accepted language for addressing parts of an XML document. It is often used in a host language such as XQuery and XSLT. However, it also serves a stand-alone query language for XML. Methods for efficient evaluation of XPath queries benefit not only XPath query engines, but also systems for more powerful languages (e.g., XQuery) which incorporated XPath.

The cited Peng portion page 432 fig. 1 example 1 reads:

I. <root>

2.<pub>

3. <bookid="I">

4. <price> 12.00 </price>

5. <name> First </name>

6. <author> A </author>

7. <price type="discount"> 10.00 </price>

8. </book>

9. <book id="2">

10. <price> 14.00 </price>

II. <name> Second </name>

12. <author> A </author>

13. <author> B </author>

14. <price type="discount"> 12.00 </price>

15. </book>

16. <year> 2002</year>

17</pub>

Figure 1: Example 1

The cited Peng portion page 432 last paragraph reads:

> As suggested by the example, we need to solve the following problems in order to evaluate even this relatively simple query. First, we may encounter data that is potentially in the result before we encounter the items required to evaluate the predicates to decide its membership. We need to buffer the potential result items. Second, items in the buffer have to be marked separately so that, after the evaluation of a predicate, we can process only the items that are affected by the predicate. Third, we have to encode the logic of the predicates in the automaton. In the above example, only when all the price children fail to satisfy the predicate (and we reach the end of the book element) does the book element fail to satisfy the predicate. In the mean time, if one of the children satisfies the predicate, we should know that the predicate is true and perform the operations accordingly. Finally, predicates access different portions of the data. Some should be evaluated when the begin

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being taken with the alleged equivalencies of the elements of claim 1 and Peng. Firstly, a review of Peng is not concerned with, does not allude to or anticipate and fails to "discloses an extensible-markup-language Path Language (XPath) evaluating method for evaluating the XPath relevant to an extensible- markup-language (XML) document by use of a computer." The first sentence of the second paragraph of Peng reads, "[W]e address the problem of evaluating XPath queries over streaming XML [23]. Claims 1-20 are not concerned with, does not allude to or anticipate "evaluating XPath queries."

Claim 1 is an XPath evaluating method for evaluating an XPath while subjecting an XML document to streaming processing. Peng does 'evaluating', but does not allude and does not present an 'evaluating method'. Claim 1 reads:

> 1. (Currently amended) An extensible-markup-language Path Language (XPath) evaluating method comprising evaluating the XPath relevant to an extensible-markup-language (XML) document by use of a computer, said step of evaluating being carried out individually concerning inputted XML events, while subjecting the XML document to streaming processing, the step of evaluating XPath comprising:
>
> a first step of serially inputting XML event strings constituting an XML document to be processed;
>
> a second step of serially evaluating the XPath respectively relevant to the inputted XML events while subjecting the XML document to streaming processing and retaining information concerning a result of partial evaluation of the XPath in given storing means when the XPath is partially established with respect to a given XML event;
>
> a third step of repeating the partial evaluation of the XPath along with the input of the XML event strings while considering the result of the partial evaluation retained in the storing means and evaluating that the XPath is established with respect to the XML document when the last part of the XPath is established; and
>
> judging establishment of the entire XPath while accumulating results of said partial evaluation enabling evaluation of the XPath by use of said streaming processing.

As stated previously, Peng does not allude to or anticipate "evaluating the XPath relevant to an extensible-markup-language (XML) document."

The office communication continues:

> *a first step of serially inputting XML event strings constituting an XML document to be processed (page 433 1st and 2nd paragraph of [2.1], 1st paragraph of [3.1], "accepts XML streams" which is "sequence of SAX events");*

to be processed (page 433 1st and 2nd paragraph of [2.1], 1st paragraph of [3.1], "accepts XML streams" which is "sequence of SAX events");

The cited Peng portion page 433 1st and 2nd paragraph of [2.1] reads:

2.1 Data Model for XML Streams

Parsers based on the SAX API process an XML document and generate a sequence of SAX events. For each opening (and closing) tag of an element, the SAX parser generates a *begin* (respectively, *end)* event. The begin event of an element comes with a list of (attribute name, attribute value) pairs with the attribute name as the key. For text contents enclosed by the opening and closing tag, the SAX parser generates a *text* event.

The streaming XML data is modeled as a sequence of SAX events, extended with the depth of the event. That is, an XML stream is a sequence {e1,e2,...e_1,...} where e_1EBUTUE:

The cited Peng portion page 433 1st paragraph of [3.1] reads:

3.1 A Simple PDA for XML **Streams**

First we introduce a FDA that accepts XML streams that have certain string. Figure 4(a) shows the state transition diagram of a FDA that accepts the XML stream in Figure 1. Text events that are not shown in the diagram map to self-transitions.

applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. A review of the referenced portions fail to show the alleged equivalency. Pang, page 433, 2nd paragraph of [2.1] reads:

"The streaming XML data is modeled as a sequence of SAX events, extended with the depth of the event. That is, an XML stream is a sequence {ei,e_2,...e_1,...}wheree_1cBuTuE."

Pang, page 433, 1st paragraph of [3.1], *reads:*

"First we introduce a PDA that accepts XML streams that have certain string. Figure 4(a) shows the state transition diagram of a PDA that accepts the XML stream in Figure 1. Text events that are not shown in the diagram map to self-transitions."

These citations are not anticipation of a step of "serially inputting XML event strings constituting an XML document to be processed."

The office communication continues:

.a second step of serially evaluating the XPath respectively relevant to the inputted XML events while subjecting the XML document to streaming processing (page 432 fig.1 and example 1, page 433 [2.2] and [3.]) and retaining information concerning a result of partial evaluation of the XPath in given storing means when the XPath is partially established with respect to a given XML event (last paragraph of page 432);

The cited Peng portion page 432 fig.1 and example 1 reads as stated above.

The cited Peng portion page 433 [2.2] and [3.] reads:

2.2 XPath

As noted earlier, XSQ implements all of XPath 1.0 [23] (including closures, aggregations, and multiple predicates) except reverse axes (such as preceding-sibling) and position functions (such as pos[0] and last 0). For the rest of this paper, we will focus on the core subset of XPath described by the grammar shown in Figure 3. An XPath query is in the form of N_1 N_2... N_n/O, which consists of a location path, N_1N_2 .. .$N_{,,,}$ and an output expression *0.* An element matches the location path if the path from the document root to that element matches the sequence of labels in the location path, and satisfies all predicates (specified syntactically using square brackets). For each matching element, the result of applying the output function to the element is added to the query result. The output expression can specify an attribute of the element, or its text value. It may also be an aggregation function (e.g., sumO) applied to the element's content. If no output expression is specified in the query, the query returns all the elements in the result set.

3. BASIC PUSHDOWN TRANSDUCER

A pushdown transducer (PDT) is a pushdown automaton (PDA) with actions defined along with the transition arcs on the automaton. It has a finite set of states which includes a start state and a set of final states, a set of input symbols, and a set of stack symbols. At each step, it fetches an input symbol from the input sequence. Based on the input symbol and the symbols in the stack, it changes the current state and operates the stack according

> to the transition function. Besides the state transition and stack operation, the transition function also defines an output operation which could generate some output during the transition. Note that traditional PDTs do not have an extra buffer and the operations for the buffer. However, as discussed in Section 1, evaluating XPath queries over XML streams requires buffering potential results.

The cited Peng portion last paragraph of page 432 reads as stated above.

applicants respectfully state that a review of the copied portions of Peng supports exception being taken with the office communication statement. A review of the referenced portions fail to show the alleged equivalency. Pang, page 433, 1st of [3. reads:

"A pushdown transducer (PDT) is a pushdown automaton (PDA) with actions defined along with the transition arcs on the automaton. It has a finite set of states which includes a start state and a set of final states, a set of input symbols, and a set of stack symbols. At each step, it fetches an input symbol from the input sequence. Based on the input symbol and the symbols in the stack, it changes the current state and operates the stack according to the transition function. Besides the state transition and stack operation, the transition function also defines an output operation which could generate some output during the transition. Note that traditional PDTs do not have an extra buffer and the operations for the buffer. However, as discussed in Section 1, evaluating XPath queries over XML streams requires buffering potential results.

Also, Example 1, makes no allusion to the second step "serially evaluating the XPath respectively relevant to the inputted XML events while subjecting the XML document to streaming processing and retaining information concerning a result of partial evaluation of the XPath in given storing means when the XPath is partially established with respect to a given XML event."

Peng is not concerned with, does not allude to or anticipate "serially evaluating' anything. Peng is not concerned with, does not allude to or anticipate "inputted XML events while subjecting the XML document to streaming processing."

Peng is not concerned with, does not allude to or anticipate "retaining information concerning a result of partial evaluation of the XPath."

Peng is not concerned when any "XPath is partially established with respect to a given XML event. Peng is not concerned with, does not allude to or anticipate the 2nd step.

Thus, Peng fails to anticipate this step either.

The office communication continues:

> *a third step of repeating the partial evaluation of the XPath along with the input of the XML event strings while considering the result of the partial evaluation retained in the storing means and evaluating that the XPath is established with respect to the XML document when the last part of the XPath is established (example 1 and last paragraph of page 432); and*

The cited Peng portion page 432 example 1 and last paragraph reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. A review of the referenced portions fail to show the alleged equivalency. Pang, example 1, starts with and reads:

EXAMPLE 1. *Consider the following query for the XML data in Figure 1: /pvMryear=2OO2J/bookrpriceC1J/at~thor.* This is not related to the this 3rd step.

Pang, last paragraph of page 432, reads:

"As suggested by the example, we need to solve the following problems in order to evaluate even this relatively simple query. First, we may encounter data that is potentially in the result before we encounter the items required to evaluate the predicates to decide its membership. We need to buffer the potential result items. Second, items in the buffer have to be marked separately so that, after the evaluation of a predicate, we can process only the items that are affected by the

predicate. Third, we have to encode the logic of the predicates in the automaton. In the above example, only when all the price children fail to satisfy the predicate (and we reach the end of the book element) does the book element fail to satisfy the predicate. In the mean time, if one of the children satisfies the predicate, we should know that the predicate is true and perform the operations accordingly. Finally, predicates access different portions of the data. Some should be evaluated when the begin tag is encountered, while others should be evaluated upon encountering the text content. There are other forms of predicates, which will be ... "

So, neither of referenced Pang portions are concerned with, do not allude to and do not anticipate this third step. Peng is not concerned with, does not allude to or anticipate any "partial evaluation."

Peng is not concerned with, does not allude to or anticipate "evaluation of the XPath along with the input of the XML event strings."

Peng is not concerned with, does not allude to or anticipate "considering the result of the partial evaluation retained in the storing means."

Peng is not concerned with, does not allude to or anticipate "evaluating that the XPath is established with respect to the XML document when the last part of the XPath is established."

Peng is certainly not concerned with, does not allude to or anticipate a "step of repeating the partial evaluation of the XPath along with the input of the XML event strings while considering the result of the partial evaluation retained in the storing means and evaluating that the XPath is established with respect to the XML document when the last part of the XPath is established.

Peng certainly doesn't anticipate a step of evaluating being carried out individually concerning inputted XML events, while subjecting the XML document to streaming processing.

judging establishment of the entire XPath while accumulating results of said partial evaluation enabling evaluation of the XPath by use of said streaming processing (page 432 fig. 1 and example 1).

The cited Peng portion page 432 fig. 1 and example 1 reads as stated above.

In response, the Applicants respectfully state that here also Peng fails to anticipate a step of judging establishment of the entire XPath while accumulating results of said partial evaluation enabling evaluation of the XPath by use of said streaming processing, as in amended claim 1.

Thus Peng is not concerned with, does not allude to or anticipate the steps of claim 1, and claim 1 and all claims that depend on claim 1 are allowable over the cited art.

***Claim 2** is rejected for the reasons set forth herein above for claim 1 and furthermore Peng teaches the XPath evaluating method wherein the second step includes the steps of:*

generating an automaton for expressing the XPath to be evaluated (page 432, 6 th paragraph of column 1, "generate the hierarchical pushdown automaton corresponding to an XPath query"); and

The cited Peng portion page 432, 6 th paragraph of column 1 reads:

The rest of this paper is organized as follows. In the rest of this section, we use examples to highlight some of the difficulties in evaluating XPath queries over XML streams. Some preliminaries, including the SAX data model and the XPath language, are covered in Section 2. The design of a basic pushdown transducer (BPDT), which corresponds to an XPath location step, is presented in Section 3. Section 4 describes our method for composing BPDTs to **generate the hierarchical pushdown automaton (HPDT) corresponding to an XPath query**. Related work is summarized Section 5. Section 6 presents some results from our empirical study of XSQ and related systems. We conclude in Section 7.

The Office Communication continues.

evaluating the XPath partially by allowing transition of a state of the automaton based on inputted respective XML events and retaining a result of the partial evaluation

as the state of the automaton (page 432 fig. 1 and example 1, page 433 1st paragraph of [3.]).

The cited Peng portion page 432 fig. 1 and example 1 reads as stated above.

The cited Peng portion page 433 1st paragraph of [3.] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. Firstly, it was shown that Peng is certainly not concerned with, does not allude to or anticipate Claim 1. Peng also is also not concerned with, does not allude to or anticipate Claim 2, which reads:

2. The XPath evaluating method according to claim 1,

wherein the second step includes the steps of:

generating an automaton for expressing the XPath to be evaluated; and

evaluating the XPath partially by allowing transition of a state of the automaton based on inputted respective XML events and retaining a result of the partial evaluation as the state of the automaton.

Peng does not have an Xpath evaluating method. A review of the referenced portions fail to show the alleged equivalency. Pang, (page 432, 6th paragraph of column 1, apparently does not allude to "generate the hierarchical pushdown automaton corresponding to an XPath query." Pang reads as above. This doesn't mention 'automaton' or "generating an automaton for expressing the XPath to be evaluated."

Pang (page 433, 1st paragraph of [3.]), copied above, does not allude to or anticipate a step of "evaluating the Xpath partially by allowing transition of a state of the automaton based on inputted respective XML events and retaining a result of the partial evaluation as the state of the automaton." Thus claim 2 is allowable over Peng for itself and because it depends on claim 1.

***Claim 3** is rejected for the reasons set forth herein above for claim 1 and furthermore Peng discloses the XPath evaluating method wherein the second step includes the steps of:*

generating a first stack which expresses the XPath to be evaluated with a string of stack elements (page 433 1st paragraph of [3.], "a set of stack symbols"); and

The cited Peng portion page 433 1st paragraph of [3.] reads as stated above.

> *generating a second stack for analyzing a nested structure of the XML document ("a set of input symbols") to be processed based on respective inputted XML events and then evaluating the XPath partially by comparing the first stack with the second stack (page 433, 1th paragraph of [3.], lines 5-10).*

The cited Peng portion page 433 1st paragraph of [3.] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. Firstly, it was shown that Peng is certainly not concerned with, does not allude to or anticipate Claim 1. Peng also is also not concerned with, does not allude to or anticipate Claim 3. A review of the referenced portions fail to show the alleged equivalency.

Peng is not concerned with, does not allude to or anticipate a step of "generating a first stack which expresses the XPath to be evaluated with a string of stack elements."

Peng is not concerned with, does not allude to or anticipate a step of "generating a second stack for analyzing a nested structure of the XML document to be processed based on respective inputted XML events and then evaluating the XPath partially by comparing the first stack with the second stack."

Thus claim 3 is allowable over Peng for itself and because it depends on claim 1.

> ***Claim 4** is rejected for the reasons set forth herein above for claim 1 and furthermore Peng teaches the XPath evaluating method wherein the second step includes the steps of:*
>
> *serially constructing a document tree indicating a document structure of the XML document to be processed based on input of respective XML events (page 436, 1st paragraph of [4.], "hierarchical pushdown transducer (HPDT), in form of a binary tree"; figure l1, and 1st paragraph of [4.2], build an HPDT from an XPath query); and*

The cited Peng portion page 436, 1st paragraph of [4.], reads:

> **4. HIERARCHICAL PDT**

The BPDTs are combined into one hierarchical pushdown transducer (HPDT), in the form of a binary tree, to process XPath queries. The key idea is to use the position of the BPDT in the HPDT to encode the results of all predicates. The BFDT can determine

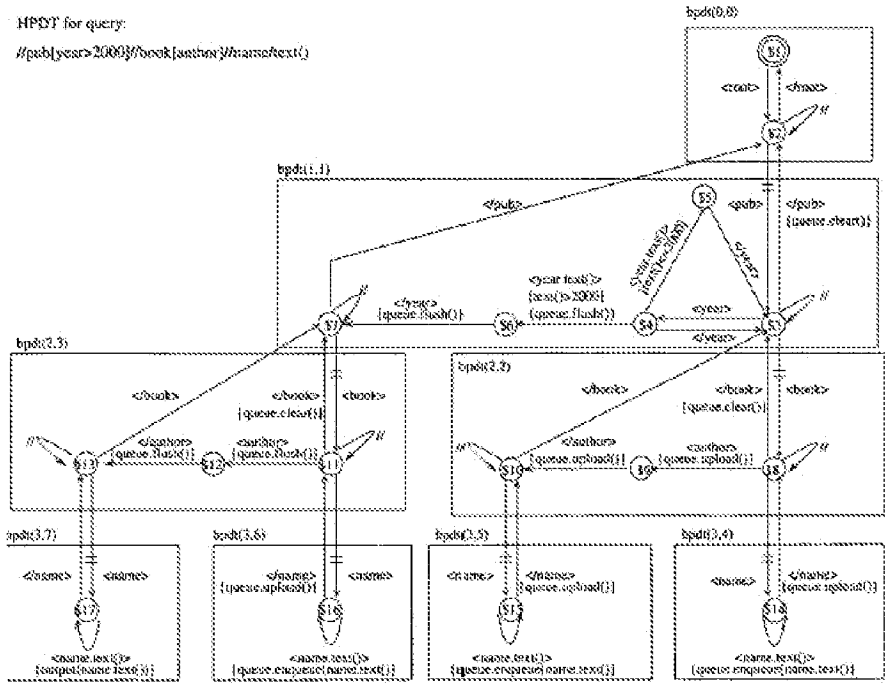

Figure 11: HPDT generated for query: //pub[year>2000]//book[author]//name/text()

whether a predicate has been evaluated or not by its own position, which is fixed and easy to get in a binary tree. Therefore, the buffer operations in the BPDTs can be determined accordingly. Due to space limitation, we only give a brief description of the algorithms in this section.

evaluating the XPath along with construction of the document tree by use of the document tree including a part which has been constructed (figures 5-11, and example 5, page 436, 1st of [4.2]).

The cited Peng portion figure 11 reads:

The cited Peng portion 1st paragraph of [4.2] reads:

4.2 Building HPDT from XPath Queries

We now describe how to build an HPDT from an XPath query. Since the BFDT decides its own buffer operation based on its position in the HPDT, we denote the position of each BPDT by a unique ID *(l,k),* where $l \geq 0$ is the depth for the BPDT in the HPDT system and $k \geq 0$ is its sequence number within the layer (right to left). Given an XPath query $N_1N_2...N_n/O$, the BPDTs,

together with the IDs, are generated as described as follows. We first generate a root BPDT as in Figure 12. The root BPDT is used to consume the *<root>*, *</root>* events, which are generated by the SAX parser for the document root for every XML document. The root BFDT has an ID (0,0). For location step N_i, we go through all the BFDTs *bpdt (i — I, k),* which are generated from $N_{i,-1}$, (N_0 could be thought as */root).* For each existing *bpdt(i — l,k),* if it has an NA state, we generate a *bpdt(i,2k)* as its right child, which use the NA state of *bpdt(i—* l,k) as its START state. If *bpdt(i—* l,k) does not have an NA state, we set *bpdt(i,2k)* to NULL. Similarly, we generate a *bpdt(i,2k+* 1) as the left child of *bpdt(i— l,k),* which uses the TRUE state *of bpdt(i— l,k)* as its START state.

'For details, please see our longer version of technical report at www.ca .umd.edu/pengfeng/xsq<root>

The cited Peng Example 5 reads:

EXAMPLE 5. The HPDT starts from state $1. It follows the rule as the usual PDT When it encounters the name "first ", it is in state $14, thus it en queues the text content "first" into the buffer of bpdt(3,4). At the end event of the name element, the item is uploaded to the buffer of bpdt(2,2). The next event is begin event of the author element, thus the HPDT goes from state $8 to state $9 and uploads the item to the buffer of bpdt(1, 1). The same process applies to the item "second", which is the name element of the second book. Then at the begin event of the year element, the HPDT is in state $3 and the buffer of bpdt (1, 1) contains two items: "first" and "second". When the HPDT encounters the text event of the year element, it evaluates the predicate (year.text() > 2000]. The result is true. Thus the HPDT goes from state $4 to $6 and flush the content in its buffer to the output. Therefore, the HPDT returns the right result for the query. From this example, we see that in each BPDT the buffer operations can be determined based on its position in the HPDT For example, for bpdt(3,4), we know it is the right child of bpdt(2, 2). This fact indicates it is connected to the NA state of bpdt (2,2). Thus, when the HPDT reaches the bpdt (3,4), the predicate in bpdt (2,2) has not been evaluated yet. Similarly, since bpdt(2, 2) is the right child of bpdt (1, 1), we know that when the HPDT reaches bpdt(2, 2), the predicate in bpdt(1, 1) has not been evaluated yet. Combine these facts, when the HPDT

is in bpdt(3,4), we know that both predicates have not been evaluated yet. Notice that these information can be obtained solely from the positions of the BPDTs, it is easy to determine the buffer operations in the BPDTs systematically. The details are described below.

The cited Peng portion 1st of [4.2] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with the above office communication statements. Firstly, it was shown that Peng is certainly not concerned with, does not allude to or anticipate Claim 1, or any partial evaluation of an XPath. Peng also is also not concerned with, does not allude to or anticipate Claim 4. A review of the referenced portions fail to show the alleged equivalency.

Peng is not concerned with, does not allude to or anticipate a step of "serially constructing a document tree indicating a document structure of the XML document to be processed based on input of respective XML events."

Peng is not concerned with, does not allude to or anticipate a step of "evaluating the XPath along with construction of the document tree by use of the document tree including a part which has been constructed.

Thus claim 4 is allowable over Peng for itself and because it depends on claim 1. Claim 4 is further narrowed in a manner that is certainly allowable of Peng.

The Office Communication continues.

> ***With respect to claim 5,*** *Peng discloses an XPath evaluating apparatus comprising:*
>
> *an evaluation executing unit being embodied in a tangible computer readable medium, and employed for inputting XML event strings constituting an XML document and serially evaluating the XPath with respect to each of XML events while subjecting the XML document to streaming processing, said serially evaluating being carried out individually concerning inputted XML events, while subjecting the XML document to*

streaming processing and while retaining information concerning a result of partial evaluation of the XPath when the XPath is partially established with respect to a given XML event, and evaluating that the XPath is established with respect to the XML document when the last step of the XPath is established (page 433, 1st of [3], "Basic Pushdown Transducer", page 432 example 1); and

The cited Peng portion page 432 example 1 reads as stated above.

EXAMPLE 1. Consider the following query for the XML data in Figure 1: /pub(year-2OO2]/book(price-11]/author. When we encounter the first author element in the stream, we know that it satisfies the path Ipub/b 00k/author. However, the predicate in the first location step, (year 2002), cannot be evaluated yet, since we have not encountered all the year sub elements. We have encountered the first prtce sub element of the book element. However we cannot determine whether the book fails the predicate (price < 11], since there may be more price sub elements. Therefore we need to buffer the book element. When we encounter the second price element of the book, the second predicate evaluates to true. Since we still do not know the year of the pub element, the author A must continue to be buffered. When we encounter the two author sub elements of the second book, we need to buffer the authors A and B as well. Now there are two As and one B in the buffer Next we encounter the second price element of the second book, and it does not satisfy the predicate. When we reach the end of the second book element, we know that the predicate (price < 11] evaluates to false, since there are no more price sub elements. Thus, the two author elements of the second book should be removed from the buffer Note that one author, A, is still in the buffer since it belongs to the first book. Later, we determine that the year element of the pub element satisfies the first predicate. By noting that the author A in the buffer has already satisfied the other predicate, we determine that the author A should be sent to the output immediately.

The Office Communication continues.

an XML event transferring unit being embodied in a tangible computer readable medium, and employed for inputting the XML event strings constituting the XML document to be processed and serially transferring the XML event strings to the evaluation executing unit (page 433, [2.1], "SAX parser"); and

The cited Peng portion page 433, [2.1], "SAX parser") reads as stated above.

a judging unit judging establishment of the entire XPath while accumulating results of said partial evaluation enabling evaluation of the XPath by use of said streaming processing (figures 5-11, and example 5, page 436, 1st of [4.2]).

The cited Peng portion figures 5-11 reads as stated above.

The cited Peng Example 5 reads as stated above.

The cited Peng portion 1st paragraph of [4.2] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception is also taken with this office communication statement. Firstly, it was shown that Peng is certainly not concerned with, does not allude to or anticipate Claim 1, or any partial evaluation of an XPath. Peng also is also not concerned with, does not allude to or anticipate Claim 5. Claim 5 is an apparatus claim equivalent of method claim 1. As with claim 1, a review of the referenced portions fail to show the alleged equivalency.

Peng is not concerned with, does not allude to or anticipate an element of "an evaluation executing unit" or an XML event transferring unit as in claim 5. Thus claim 5 is allowable over Peng.

***Claim 6** is rejected for the reasons set forth herein above for claim 5 and furthermore Peng teaches the XPath evaluating apparatus, further comprising:*

an automaton generating unit being embodied in a tangible computer readable medium, and employed for generating an automaton which expresses the XPath to be evaluated (page 433 1st of [3.], "pushdown transducer"),

The cited Peng page 433 1st of [3.] Reads as stated above.

wherein the evaluation executing unit performs partial evaluation of the XPath by allowing a state of the automaton generated by the automaton generating unit to perform transition based on the XML events transferred from the XML event transferring unit, and retains a result of the partial evaluation as the state of the automaton (page 433 1st paragraph of [3.]).

The cited Peng portion page 433 1st of [3.] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. Firstly, it was shown that

Peng is certainly not concerned with, does not allude to or anticipate Claim 1. Peng also is also not concerned with, does not allude to or anticipate Claims 5 or 6. A review of the referenced portions fail to show the alleged equivalency.

Peng is not concerned with, does not allude to or anticipate an element of an automaton generating unit as in claim 6.

Thus claim 6 is allowable over Peng for itself and because it depends on claim 5.

> ***Claim 7*** *is rejected for the reasons set forth herein above for claim 5 and furthermore Peng teaches the XPath evaluating apparatus, further comprising:*
>
> *a stack generating unit being embodied in a tangible computer readable medium, and employed for generating a first stack which expresses the XPath to be evaluated with a string of stack elements (page 433 1st paragraph of [3.], "a set of stack symbols"),*

The cited Peng portion page 433 1st paragraph of [3.] reads as stated above.

> *wherein the evaluation executing unit performs partial evaluation of the XPath by generating a second stack for analyzing a nested structure of the XML document subject to processing based on the XML events transferred from the XML event transferring unit and then comparing the first stack generated by the stack generating unit with the second stack (page 433 1st paragraph of [3.], [3.1]).*

The cited Peng portion page 433 1st paragraph of [3.] reads as stated above.

The cited Peng portion page 433 1st paragraph of [3.1] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. Firstly, it was shown that Peng is certainly not concerned with, does not allude to or anticipate Claim 1, or any partial evaluation of an XPath. Peng also is also not concerned with, does not allude to or anticipate Claim 7. A review of the referenced portions fail to show the alleged equivalency.

Peng is not concerned with, does not allude to or anticipate an element of a stack generating unit for generating a first stack which expresses the XPath to be evaluated with a string of stack element. Peng is not concerned with, does not allude to or anticipate an "evaluation executing

unit performs partial evaluation of the XPath by generating a second stack for analyzing a nested structure of the XML document subject to processing based on the XML events transferred from the XML event transferring unit and then comparing the first stack generated by the stack generating unit with the second stack."

Thus claim 7 is allowable over Peng for itself and because it depends on claim 5.

> ***With respect to claim 8,*** *Peng teaches an XPath evaluating apparatus comprising:*
>
> *a document tree constructing unit being embodied in a tangible computer readable medium, and employed for inputting XML event strings which constitute an XML document and serially constructing a document tree indicating a document structure of the XML document based on inputted XML events along with the input of the respective XML events while subjecting the XML document to streaming processing (page 434, [3.2], "building the BPDT" with figure 5-9);*

The cited Peng portion page 434, [3.2] reads as stated above.

The cited Peng portion figure 5-9 reads as stated above.

> *an XML event transferring unit being embodied in a tangible. computer readable medium, and employed for inputting the XML event strings which constitute the XML document to be processed and serially transferring the XML event strings to the document tree constructing unit (page 433, [2.1], "SAX parser"); and*

The cited Peng portion page 433, [2.1], "SAX parser") reads as stated above.

The cited Peng portion page 434, [3.2] reads:

> 3.2 **Building the BPDT**
>
> Our solution to this problem is based on our observations on the following example. **EXAMPLE 3.** For the following XPath query, consider the second location step (/book(author3):
>
> **Q:** `/pub(year)20003/book(authorJ/name/text 0`
>
> In a PDTfor this query, we need to perform at least three tasks for this location step: 1. If the book element does have an author subelement, we need to remember the fact for future use.

2. If the book element does not have an author subelement, we need to make sure that if the name of the current book element has been in the buffer; it is deleted from the buffer

3. If the book element does have an author subelement, we need to make sure that ~f the name of the current book has been in the buffer; ii is sent to the output if all the predicates have evaluated to true. If some of the predicates have not been evaluated, we should hold the content in the buffer and handle it later

The event upon which we can perform the first task is the begin event of the author element. The event upon which we can perform the second task is the end event of the book element since until then we cannot be sure that the book element does not have an author subelement. At the begin event of the author element, we also need to perform the third task since we know now the predicate in the current location step is true.

Intuitively, these observations suggest associating a PDT similar to the one suggested by Figure 8 with a location step of this form. (The buffer operations will be explained in detail shortly.)

Using similar analysis to the above example for the predicates used in XPath , the location steps in any XFath queries can be categorized into the following classes based on the events upon which the predicates are evaluated.

I. Test whether the current element has a specified attribute, or whether the attribute satisfies some condition, (e.g., /book [aid], fbook['~id < 10]).

2. Test whether the current element contains some text, or whether the text value satisfies some condition, (e.g., /year[text() =2000]).

3. Test whether the current element has a specified type of child, (e.g., /book [author]).

4. Test whether the current element's specified child contains an attribute, or whether the value of the attribute satisfies some condition, (e.g., /pub[book-id 5-10]).

5. Test whether the specified child of the current element has a value that satisfies some condition, (e.g., /book [year = 2000]).

Based on the above categorization, we design a template for each category of location steps. Figure 5 to Figure 9 summarize these templates. In each template, there is a START state, a TRUE state that indicates the predicate in this location step has evaluated to true, and an NA state that indicates the predicate has not yet been evaluated. The PDT generated from a location step using the template is called a basic pushdown automaton (BPDT). The BPDT has two important features:

I. The result of the predicate is encoded in the states. It is easy to show that whenever the BPDT is in the TRUE state, the predicate has evaluated to true; whenever the BPDT is in the NA state, the predicate has not yet been evaluated.

2. The logic of the predicate is encoded in the BFDT. For example, in Figure 9, we can see the exact logic we want for location steps such as /pub [year2002] in Example I. If one of the children satisfies the criterion, the BPDT will move to the TRUE state. Only if all the children fail the predicate does the BFDT return to the START state from the NA state, signifying that the predicate has evaluated to false.

an evaluation executing unit being embodied in a tangible computer readable medium, and employed for evaluating the XPath along with construction of the document tree by the document tree constructing unit being carried out individually concerning inputted XML events, while subjecting the XML document to said streaming processing, using the document tree with a part which has been constructed (page 436, 1st paragraph of [4.], "hierarchical pushdown transducer (HPDT), in form of a binary tree"; figure 1, and 1st paragraph of [4.2], build an HPDT from an XPath query, page 434 [3.2]); and

The cited Peng portion page 436, 1st paragraph of [4.] reads as stated above.

The cited Peng portion figure 1 reads as stated above.

The cited Peng portion 1st paragraph of [4.2] reads as stated above.

a judging unit judging establishment of the entire XPath while accumulating results of said partial evaluation enabling evaluation of the XPath by use of said streaming processing (figures 5-11, and example 5, page 436, 1st of [4.2])!

The cited Peng portion figures 5-11 reads as stated above.

The cited Peng Example 5 reads as stated above.

The cited Peng portion 1st paragraph of [4.2] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. Firstly, it was shown that Peng is certainly not concerned with, does not allude to or anticipate Claim 1, or any partial evaluation of an XPath. Peng also is also not concerned with, does not allude to or anticipate Claim 8. A review of the referenced portions fail to show the alleged equivalency.

Peng is not concerned with, does not allude to or anticipate an "XPath evaluating apparatus of claim 8. Peng is not concerned with, does not allude to or anticipate "a document tree constructing unit, for inputting XML event strings which constitute an XML document and serially constructing a document tree indicating a document structure of the XML document

based on inputted XML events along with the input of the respective XML events while subjecting the XML document to streaming processing."

Peng is not concerned with, does not allude to or anticipate "an XML event transferring unit" "employed for inputting the XML event strings which constitute the XML document to be processed and serially transferring the XML event strings to the document tree constructing unit.

Peng is not concerned with, does not allude to or anticipate the evaluation executing unit of claim 8. Thus claim 8 is allowable over Peng.

> ***Claim 9*** *is rejected for the reasons set forth herein above for claim 8 and furthermore Peng teaches the XPath evaluating apparatus, wherein the evaluation executing unit retains information concerning a result of partial evaluation of the XPath when the XPath is partially established upon the evaluation of the XPath using the document tree (page 434, [3.3], "Buffer operation in BPDT").*

The cited Peng portion page 434, [3.3] reads:

> **3.3 Buffer operations in BPDT**
>
> In contrast to the simple FDA, each BPDT has 'a buffer of its own that is organized as a queue. The operations on the buffer are as follows:
>
> 1. Q.enqueue(v): add *v* to the end of the queue;
> 2. Q.clearO: remove all the items in the queue;
> 3. Q.fiushO: send all items in the queue to the output in FIFO order;
> 4. Q.uploadO: move all the items in the queue to the end of the queue of the BPDT that is the parent of this BPDT in the HPDT network, as explained further in Section 4.
>
> Note that we do not have the *dequeue* operation for the queue since all the items in the queue will be operated on together: either to be *cleared* or to be *flushed* to output.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. Firstly, it was shown that Peng is certainly not concerned with, does not allude to or anticipate Claims 1 and 8, or any partial evaluation of an XPath. Peng also is also not concerned with, does not allude to or anticipate Claim 9. A review of the referenced portions fail to show the alleged equivalency.

Peng is not concerned with, does not allude to or anticipate a limitation "wherein the evaluation executing unit retains information concerning a result of partial evaluation of the XPath when the XPath is partially established upon the evaluation of the XPath using the document tree."

Thus claim 9 is allowable over Peng for itself and because it depends on claim 8.

With respect to claim 10, *Peng teaches an information processing apparatus comprising:.*

an XML parser for analyzing an XML document to be processed and thereby generating XML event strings (page 433, [2.1]);

The cited Peng portion page 433, [2.1] reads as stated above.

an XPath evaluating unit being embodied in a tangible computer readable medium, and employed for serially inputting the XML event strings generated by the XML parser and evaluating the XPath with respect to each of inputted XML events by streaming processing, said step of evaluating being carried out individually concerning inputted XML events, while subjecting the XML document to streaming processing (page 433 1st of [3.], "PDA", page 432 example 1); and

The cited Peng page 433 1st of [3.] Reads as stated above.

The cited Peng portion page 432 example 1 reads as stated above.

an application executing unit being embodied in a tangible computer readable medium, and employed for inputting the XML events generated by the XML parser and performing processing with respect to the XML document configured by the XML events in response to an evaluation result of the XPath by the XPath evaluating unit (page 434, [3.2], "building the BPDT"),

The cited Peng page 434, [3.2] reads as stated above.

wherein the XPath evaluating unit serially evaluates the XPath with respect to each of the XML events, retains information concerning a result of partial evaluation of the XPath when the XPath is partially established with respect to a given XML event, and judges that the XPath is established with respect to the XML document when the last step of the XPath is established (page 433 1st of [3], "Basic Pushdown Transducer"); and

The cited Peng portion page 433 1st of [3] reads as stated above..

> *a judging unit judging establishment of the entire XPath while accumulating results of said partial evaluation enabling evaluation of the XPath by use of said streaming processing (figures 5-11, and example 5, page 436, 1st of [4.2]).*

The cited Peng portion figures 5-11 reads as stated above.

The cited Peng portion Example 5 reads as stated above.

The cited Peng portion page 436, 1st paragraph of [4.2] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. Firstly, it was shown that Peng is certainly not concerned with, does not allude to or anticipate Claim 1, or any partial evaluation of an XPath. Peng also is also not concerned with, does not allude to or anticipate Claim 10. A review of the referenced portions failed to show the alleged equivalency.

Peng is not concerned with, does not allude to or anticipate a combination of elements as in claim 10. Peng does not combine and/or does not have an XML parser, an XPath evaluating unit and an application executing unit of claim 10.

Peng is not concerned with, does not allude to or anticipate an apparatus wherein an "XPath evaluating unit serially evaluates the XPath with respect to each of the XML events, retains information concerning a result of partial evaluation of the XPath when the XPath is partially established with respect to a given XML event, and judges that the XPath is established with respect to the XML document when the last step of the XPath is established."

Thus claim 10 is allowable over Peng.

> ***Claim 11*** *is rejected for the reasons set forth herein above for claim 10 and furthermore Peng teaches the information processing apparatus, wherein the XPath evaluating unit generates an automaton for expressing the XPath to be evaluated, performs partial evaluation of the XPath by allowing transition of a state of the automaton based on the XML events generated by the XML parser, and retains a result of the partial evaluation as the state of the automaton (example 1 and last paragraph of page 432, when author element in input stream is encountered, XPath is evaluated, and it satisfied the path /pub/book/author. However, book element, author elements are buffered to wait for later input stream events process).*

The cited Peng portion page 432 example 1 and last paragraph reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports exception being also taken with this office communication statement. Firstly, it was shown that Peng is certainly not concerned with, does not allude to or anticipate Claim 10, or any partial evaluation of an XPath. Peng also is also not concerned with, does not allude to or anticipate Claim 11. A review of the referenced portions fail to show the alleged equivalency.

Peng is not concerned with, does not allude to or anticipate a limitation of "wherein the XPath evaluating unit generates an automaton for expressing the XPath to be evaluated."

Peng is not concerned with, does not allude to or anticipate a limitation of "performs partial evaluation of the XPath by allowing transition of a state of the automaton based on the XML events generated by the XML parser, and retains a result of the partial evaluation as the state of the automaton."

Thus claim 11 is allowable over Peng for itself and because it depends on claim 10.

> ***Claim 12*** *is rejected for the reasons set forth herein above for claim 10 and furthermore Peng teaches the information processing apparatus, wherein the XPath evaluating unit generates a first stack which expresses the XPath to be evaluated with a string of stack elements, generates a second stack for analyzing a nested structure of the XML document to be processed based on the XML events generated by the XML parser, and performs partial evaluation of the XPath by then comparing the first stack with the second stack (page 433 1st paragraph of [3.], lines 5-10).*

The cited Peng portion page 433 1st paragraph of [3.] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports it was shown that Peng does not anticipate claim 10. Peng does not anticipate claim 12. Peng is not concerned with, does not allude to or anticipate a limitation of "wherein the XPath evaluating unit serially constructs a document tree indicating a document structure of the XML document to be processed based on inputted XML events along with the input of the respective XML events

generated by the XML parser, and evaluates the XPath by use of the document tree with a part which has been constructed."

Thus claim 12 is allowable over Peng for itself and because it depends on claim 10.

> ***Claim 13*** *is rejected for the reasons set forth herein above for claim 10 and furthermore Peng teaches the information processing apparatus, wherein the XPath evaluating unit serially constructs a document tree indicating a document structure of the XML document to be processed based on inputted XML events along with the input of the respective XML events generated by the XML parser, and evaluates the XPath by use of the document tree with a part which has been constructed (page 436, 1st paragraph of [4.], "hierarchical pushdown transducer (HPDT), in form of a binary tree"; figure 1, and 1st paragraph of [4.2], build an HPDT from an XPath query).*

The cited Peng portion figures 1 reads as stated above.

The cited Peng page 436, 1st paragraph of [4.] reads as stated above.

The cited Peng portion 1st paragraph of [4.2] reads as stated above.

In response, the applicants respectfully state that a review of the copied portions of Peng supports it was shown that Peng does not anticipate claim 10. Peng does not anticipate claim 13. Peng is not concerned with, does not allude to or anticipate a limitation of "wherein the XPath evaluating unit serially constructs a document tree indicating a document structure of the XML document to be processed based on inputted XML events along with the input of the respective XML events generated by the XML parser, and evaluates the XPath by use of the document tree with a part which has been constructed." Thus claim 13 is allowable over Peng for itself and because it depends on claim 10.

> ***Claim 14*** *is rejected on grounds corresponding to the reasons given above for claim 1. The claim 1 claims limitations of the XPath evaluating method while the claim 14 claims limitations of a program causing the computer to execute the procedure for carrying out the steps of claim 1.*

In response, the applicants respectfully state that a review of the copied portions of Peng supports indeed as with claim 1, claim 14 for a program to do the steps of claim 1 is allowable because it depends on allowable claim 1.

> ***Claim 15*** *is rejected on grounds corresponding to the reasons given above for claim 1. The claim 1 claims limitations of the XPath evaluating method while the claim 15 claims limitations of an article of manufacture comprising computer readable program code means for causing a computer to effect the steps of claim 1.*

In response, the applicants respectfully state that a review of the copied portions of Peng supports indeed as with claim 1, claim 1 for an article of manufacture to do the steps of claim 1 is allowable because it depends on allowable claim 1.

> ***Claims 16, and 17*** *are rejected on grounds corresponding to the reasons given above for claim I. The claim 1 claims limitations of the XPath evaluating method while the claims 16, 17 claim limitations of a program storage device readable by machine to perform the steps of claim 1.*

In response, the applicants respectfully state that a review of the copied portions of Peng supports indeed as with claim 1, claims 16 and 17 for a program storage device to do the steps of claim 1 is allowable because it depends on allowable claim 1.

> ***Claim 18*** *are rejected on grounds corresponding to the reasons given above for claim 5. The claim 5 claims limitations of the XPath evaluating apparatus while the claim 18 claims limitations of a computer program product for causing a computer to effect the XPath evaluating apparatus of claim 5.*

In response, the applicants respectfully state that a review of the copied portions of Peng supports indeed as with claim 5, claim 18 for a program product to do the functions of claim 5 is allowable because it depends on allowable claim 5.

> ***Claim 19*** *are rejected on grounds corresponding to the reasons given above for claim 8. The claim 8 claims limitations of the XPath evaluating apparatus while the claim 19 claims limitations of a computer program product for causing a computer to effect the XPath evaluating apparatus of claim 8.*

In response, the applicants respectfully state that a review of the copied portions of Peng supports indeed as with claim 8, claim 19 for a program product to do the functions of claim 8 is allowable because it depends on allowable claim 8.

Claim 20 *are rejected on grounds corresponding to the reasons given above for claim 10. The claim 10 claims limitations of the information processing apparatus while the claim 20 claims limitations of a computer program product for causing a computer to effect the information processing apparatus of claim 10.*

In response, the applicants respectfully state that a review of the copied portions of Peng supports indeed as with claim 10, claim 20 for a program product to do the functions of claim 10 is allowable because it depends on allowable claim 10.

Response to Argument

Response to Argument

5. Applicants' arguments regarding the 102(a) rejection based upon Peng are not persuasive. The examiner respectfully traverses applicants' arguments.

With respect to applicants argument that Peng does not anticipate "evaluating the XPath relevant to an extensible-markup-language (XML) document", the examiner refers applicants to page 431 [1.] 2nd paragraph. Peng discloses ***evaluating XPath over streaming XML.*** *Therefore, "evaluating the XPath relevant to an extensible-markup-language (XML) document" is taught in Peng's reference.*

With respect to applicant's argument that Peng does not anticipation of "serially inputting XML event strings constituting an XML document to be processed", the examiner respectfully disagrees. In page 433 [2.1], Peng teaches ***the XML document is processed and generate a sequence of events.*** *Additional, [3.1] recites* ***PDA accepts the streaming XML.*** *Therefore., "serially inputting XML event strings constituting an XML document to be processed" is taught in Peng's reference.*

With respect to applicants argument that Peng does not anticipate "serially evaluating "inputted XML events while subjecting the XML document to streaming processing", and "XPath is partially established with respect to a given XML event", the examiner respectfully disagrees. The examiner refers applicants to 1st paragraph of [3.] in page 433, as the explanation supra, the XML document is processed to generate a sequence of XML event strings. Fig. 1 depicts XML event strings, page 432 example 1: the process of evaluating the XPath query /pub[year=2002]/book[price<11]/author for the XML event strings (fig. 1). In the example each event element of XML document is evaluated along with each location step in the Xpath /pub/book/author. The location step pub with predicate [year=2002] and book with predicate [price < 11] (figure 1 of example 1) are fetched for evaluating. The fist book element event, the author element event will be buffered until the sub event elements evaluated. After the predicate [price < 11] of book location step is evaluated using the second price event element of the second book event element, the Xpath is partially established. Therefore, anticipate "serially

evaluating "inputted XML events while subjecting the XML document to streaming processing", and "XPath is partially established with respect to a given XML event" are taught in Peng's reference.

The cited Peng portion 1st paragraph of [3.] in page 433 reads:

> Such a FDA can be adapted to a filtering system for XML documents using the following method. Suppose we want to find all the documents that contain some elements that have the pattern //pub//book//name. We can just remove all the branches in Figure 4(a) to make it a filter FDA shown Figure 4(b). Note that if the state transition of the next event is not defined in the filter FDA, the filter PDA just stays in the same state. Whenever the filter FDA reaches state *$8,* we know that the current XML document contains an element that satisfies the filter expression and we can return the document to the user. Moreover, if we put output functions in state *$8* such that it can output the content of *name* element, the filter FDA becomes a PDT that can answer the XPath query.

The cited Peng portion page 432 example 1 reads as stated above.

The cited Peng portion page 432 fig. 1 and example 1 reads as stated above.

The Office Communication continues.

With respect to applicants' argument that Peng does not anticipate "step of repeating the partial evaluation of the XPath along with the input of the XML event strings while considering the result of the partial evaluation retained in the storing means and evaluating that the XPath is established with respect to the XML document when the last part of the XPath is established", the examiner respectfully disagrees. The examiner refers applicants to the explanation supra. Additionally, after the predicate [price < 11] of book location step is evaluated using the second price event element of the second book event element, the Xpath is partially established. The predicate [year=2002] of pub location path will be evaluated using year event element to complete evaluating entire XPath. During the evaluation of the Xpath, the buffer is used to store the partial evaluation until the entire Xpath is evaluated. Therefore, the recited limitation is taught in Peng's reference.

6. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

With respect to applicants' argument that "generating an automaton for expressing the XPath to be evaluated" because "page 432, 6th paragraph of column 1, apparently does not allude to ***generate the hierarchical pushdown automaton corresponding to an XPath query****', the examiner respectfully disagrees.* ***"generate the hierarchical pushdown automaton corresponding to an XPath query(*** *is in page 432, 6Th paragraph of column 1. Therefore, "generating an automaton for expressing the XPath to be evaluated" is taught in Peng's reference.*

The cited Peng portion page 432, 6th paragraph of column 1 reads:

> The rest of this paper is organized as follows. In the rest of this section, we use examples to highlight some of the difficulties in evaluating XPath queries over XML streams. Some preliminaries, including the SAX data model and the XPath language, are covered in Section 2. The design of a basic pushdown transducer (BPDT), which corresponds to an XPath location step, is presented in Section 3. Section 4 describes our method for composing BPDTs to generate the hierarchical pushdown automaton (HPDT) corresponding to an XPath query. Related work is summarized Section *5.* Section 6 presents some results from our empirical study of XSQ and related systems. We conclude in Section 7.

The Office Communication continues.

With respect to applicants' argument that Peng does not anticipate a step of "evaluating the XPath partially by allowing transition of a state of the automaton based on inputted respective XML events and retaining a result of the partial evaluation as the state of the automaton", the examiner respectfully disagrees. page 433 1st paragraph of [3.], Peng teaches for each input event from the input sequence (figure 1 of example 1) is fetched ,the current state of PDT is changed and buffered the result. Therefore, the limitation is taught in Peng's reference.

With respect to applicants' argument that Peng does not anticipate a step of "generating a second stack for analyzing a nested structure of the XML document to be processed based on respective inputted XML events and then evaluating the XPath partially by comparing the first stack with the second stack", the examiner respectfully disagrees. In 1st paragraph of [3.] page 433, "a set of input symbol" (fig.1) is represent for "a nested structure of the XML document" as in claim limitation, and "the set of stack input symbols". Additionally, the example 1 discloses the comparing each of events in the document to the each element in the XPath. Therefore, the limitation is taught in Peng's reference.

With respect to applicants argument regarding to claim 4 that Peng does not anticipate a step of "serially constructing a document tree indicating a document structure of the XML document to be processed based on input of respective XML events"; and "evaluating the XPath along with construction of the document tree by use

of the document tree including a part which has been constructed". ***The examiner recites in the office action*** *"serially constructing a document tree indicating a document structure of the XML document to be processed based on input of respective XML events (page 436, 1st paragraph of [4.], "hierarchical pushdown transducer (HPDT), in form of a binary tree"; figure 1, and 1st paragraph of [4.2], build an HPDT from an XPath query); and evaluating the XPath along with construction of the document tree by use of the document tree including a part which has been constructed (figures 5-11, and example 5, page 436, 1st of [4.2])".*

With respect to applicants argument regarding to claim 5 that Peng does not anticipate an element of "an evaluation executing unit", ***the examiner recites in the office action*** *"an evaluation executing unit.... (page 433 1st of [3], "Basic Pushdown Transducer")".*

With respect to applicants argument regarding to claim 6 that Peng does not anticipate an element of "an automaton generating unit", ***the examiner recites in the office action*** *"an automaton generating unit (page 433, 1st of [3.], "pushdown transducer")".*

With respect to applicants argument regarding to claim 7 that Peng does not anticipate an element of "a stack generating unit being embodied in a tangible computer readable medium, and employed for generating a first stack which expresses the XPath to be evaluated with a string of stack elements", and "the evaluation executing unit performs partial evaluation of the XPath by generating a second stack for analyzing a nested structure of the XML document subject to processing based on the XML events transferred from the XML event transferring unit and then comparing the first stack generated by the stack generating unit with the second stack". ***The examiner recites in the office action*** *"a stack generating unit being embodied in a tangible computer readable medium, and employed for generating a first stack which expresses the XPath to be evaluated with a string of stack elements (page 433 1st paragraph of [3.], "a set of stack symbols"), wherein the evaluation executing unit performs partial evaluation of the XPath by generating a second stack for analyzing a nested structure of the XML document subject to processing based on the XML events transferred from the XML event transferring unit and then comparing the first stack generated by the stack generating unit with the second stack (page 433 1st paragraph of [3.], [3.1])."*

With respect to applicants argument regarding to claim 8 that Peng does not anticipate "a document tree constructing unit being embodied in a tangible computer readable medium, and employed for inputting XML event strings which constitute an XML document and serially constructing a document tree indicating a document structure of the XML document based on inputted XML events along with the input of the respective XML events while subjecting the XML document to streaming processing; an XML event transferring unit being embodied in a tangible computer readable medium, and employed for inputting the XML event strings which constitute the XML document to be processed and serially transferring the XML event strings to the document tree

constructing unit. ***The examiner recites in the office action*** *"a document tree constructing unit being embodied in a tangible computer readable medium, and employed for inputting XML event strings which constitute an XML document and serially constructing a document tree indicating a document structure of the XML document based on inputted XML events along with the input of the respective XML events while subjecting the XML document to streaming processing (page 434, [3.2], "building the BPDT" with figure 5-9); an XML event transferring unit being embodied in a tangible computer readable medium, and employed for inputting the XML event strings which constitute the XML document to be processed and serially transferring the XML event strings to the document tree constructing unit (page 433, [2.1], "SAX parser")".*

With respect to applicants argument regarding to claim 9 that Peng does not anticipate an element of "wherein the evaluation executing unit retains information concerning a result of partial evaluation of the XPath when the XPath is partially established upon the evaluation of the XPath using the document tree", ***the examiner recites in the office action*** *"wherein the evaluation executing unit retains information concerning a result Of partial evaluation of the XPath when the XPath is partially established upon the evaluation of the XPath using the document tree (page 434, [3.3], "Buffer operation in BPDT")".*

With respect to applicants argument regarding to claim 10 that Peng does not anticipate "the XPath evaluating unit serially evaluates the XPath with respect to each of the XML events, retains information concerning a result of partial evaluation of the XPath when the XPath is partially established with respect to a given XML event, and judges that the XPath is established with respect to the XML document when the last step of the XPath is established", ***the examiner recites in the office action*** *"the XPath evaluating unit serially evaluates the XPath with respect to each of the XML events, retains information concerning a result of partial evaluation of the XPath when the XPath is partially established with respect to a given XML event, and judges that the XPath is established with respect to the XML document when the last step of the XPath is established (page 433, 1st of [3], "Basic Pushdown Transducer")".*

With respect to applicants argument regarding to claim 11 that Peng does not anticipate "the XPath evaluating unit generates an automaton for expressing the XPath to be evaluated, performs partial evaluation of the XPath by allowing transition of a state of the automaton based on the XML events generated by the XML parser, and retains a result of the partial evaluation as the state of the automaton", ***the examiner recites in the office action*** *"the XPath evaluating unit generates an automaton for expressing the XPath to be evaluated, performs partial evaluation of the XPath by allowing transition of a state of the automaton based on the XML events generated by the XML parser, and retains a result of the partial evaluation as the state of the automaton (example I and last paragraph of page 432, when author element in input stream is encountered, XPath is evaluated, and it satisfied the path /pub/book/author. However, book element, author elements are buffered to wait for later input stream events process)".*

> *With respect to applicants argument regarding to claim 12 that Peng does not anticipate "the XPath evaluating unit serially constructs a document tree indicating a document structure of the XML document to be processed based on inputted XML events along with the input of the respective XML events generated by the XML parser, and evaluates the XPath by use of the document tree with a part which has been constructed". The examiner respectfully disagrees.* ***This limitation is not recited in claim 12.***

In response the applicants respectfully state that claim 12 is amended to recite this limitation.

> *With respect to applicants argument regarding to claim 13 that Peng does not anticipate "the XPath evaluating unit serially constructs a document tree indicating a document structure of the XML document to be processed based on inputted XML events along with the input of the respective XML events generated by the XML parser, and evaluates the XPath by use of the document tree with a part which has been constructed",* ***the examiner recites in the office action*** *"wherein the XPath evaluating unit serially constructs a document tree indicating a document structure of the XML document to be processed based on inputted XML events along with the input of the respective XML events generated by the XML parser, and evaluates the XPath by use of the document tree with a part which has been constructed (page 436, 1st paragraph of [4.], "hierarchical pushdown transducer (HPDT), in form of a binary tree"; figure l1, and 1st paragraph of [4.2], build an HPDT from an XPath query)".*

In response the applicants respectfully state that a review of the copied portions of Peng supports indeed they maintain the arguments made previous differentiate over Peng. The office communication appears to selectively take parts of the claims and parts of Peng to override the arguments.

Applicants respectfully state that a review and copying of the cited portion should certainly be viewed as showing the validity of applicants arguments. Peng fails to do the all the steps and/or functions of the method and apparatus claims. The Examiner appears to be using personal knowledge in providing interpretation of the cited portions of Peng to be a teaching of each respective claim, where the interpretation is apparently using hindsight which is not allowed in a 102 rejection.

However, in order to bring this application to allowance quickly, the independent claims were narrowed. This should certainly convince the Examiner of the novelty of the invention as claimed.

Claim 12 is amended and should be allowable.

Peng may do similar things but fails to teach the steps of the present methods, and fails to teach the functions of the present apparatus claims.

It is therefore again anticipated that this amendment shows that claims 1-20 are allowable.

If the Examiner chooses to allow any claim or claims, Examiner is authorized to cancel all the non-allowed claims in order to bring allowable subject matter to issue. If any question remains, please contact the undersigned.

Please charge any fee necessary to enter this paper to deposit account 50-0510.

Respectfully submitted,

By: _____/Louis Herzberg/_____
Dr. Louis P. Herzberg
Reg. No. 41,500
Voice Tel. (845) 352-3194
Fax. (845) 352-3194

3 Cloverdale Lane
Monsey, NY 10952

Customer Number: 54856